RESEARCHERID
THOMSON REUTERS

[ICRTCET-2018]

ISSN 2348 – 8034
Impact Factor- 5.070

GLOBAL JOURNAL OF ENGINEERING SCIENCE AND RESEARCHES

PREDICTING EMPLOYEE ATTRITION FROM AN UNBALANCED HUMAN RESOURCES DATA SET USING MACHINE LEARNING

Ashwith Atluri[*1] **& Dr. Sharvani G.S**[2]
[*1]R.V College of Engineering, Bangalore, India
[2]Dept. of CSE, R.V College of Engineering, Bangalore, India

ABSTRACT

Employee attrition is something companies constantly look forward to reducing by retaining their best employees through ensuring an employee friendly atmosphere along with providing quality work and employee benefits. The purpose of this project to predict whether an employee will leave an organization given a specific set of work related parameters. Also, the factors responsible for employee attrition are highlighted to show problematic areas for rectification.

For this specific project, an unbalanced human resource data set was downloaded from Kaggle. To ensure that the prediction of employee attrition wasn't biased towards a specific side by virtue of imbalance, the dataset was balanced using a specific method mentioned later in the paper. Data was normalized on a scale of 0-1 and textual features were converted into numeric features using One Hot Encoding. Multiple machine learning models were applied to the dataset to predict attrition and the results were documented to see which model performed the best. The AutoML model for which no balancing and feature engineering was required was also used and its results were compared with the other models.

It was concluded that while AutoML gave a fairly accurate model with over 98% accuracy, the time taken for training is much higher than the other models. However, AutoML requires minimal feature engineering and less expertise in feature engineering if one wants to use the dataset directly. Data was also automatically balanced by AutoML. The models such as Support Vector Classifier with RBF kernel, Logistic Regression, Stochastic Gradient Descent were accurate up to a percent of 68% which wasn't satisfactory for this use case. Ada Boost and Gradient Boosting classifier gave an accuracy of up to 95%. Random Forest Classifier, Bagging Classifier and XGBoost Classifier gave an accuracy of up to 99%.

Keywords: *Employee, Attrition, Normalisation, Features, AutoML, Balancing, XGBoost, Boosting, Bagging.*

I. INTRODUCTION

One of the biggest challenge companies face on the human resources front is reducing employee attrition. One of the prime objectives is to retain top talent by ensuring that employees are paid well, provided good benefits, a comfortable working atmosphere and are given quality work. Companies try to reduce their investment in hiring by trying to retain the present workforce. A number of factors influence employee attrition which include pay, the department they work in, employee satisfaction, working hours, team dynamics and so on.

The purpose of this project to predict whether an employee will leave an organization given a specific set of work related parameters. Also, the factors responsible for employee attrition are highlighted to show problematic areas for rectification.

II. LITERATURE REVIEW

The model which requires no feature engineering and data preprocessing to be done from the user's end is AutoML [1]. It takes care of imputation of missing values, balancing the dataset, one hot encoding, data rescaling. This model has been used to predict the employee attrition in this project which requires minimal effort from the user. Support Vector Classifier which is a Support Vector Machine and is effective on high dimensional datasets is also used on the processed dataset [2]. Logistic regression, is a linear model for classification and is also known as the log-linear classifier, logit regression, or maximum-entropy classification (MaxEnt). Logistic function is used for modelling the probabilities indicating the possible outcomes of a trial [3]. Stochastic Gradient Descent (SGD) is a straightforward and economic approach of linear classifiers under convex loss functions such like Logistic Regression and SVMs (Support Vector Machines) and its advantage is efficiency [4].

Bagging methods are a class of algorithms which build several instances of a same type of estimator on subsets of the original training set randomly and then use the predictions of each one of them to form a final prediction. These are used as a way to reduce the variance of a base estimator (e.g., a decision tree), by making an ensemble by incorporating randomization into its construction procedure. The method is called Bagging [5] when samples are drawn with replacement. In random forests classifier [6], each tree in the ensemble is built from a sample drawn with replacement (a bootstrap sample) from the training set. During the construction of the tree the node is split and the split that is chosen is no longer the best split among all features. The split that is picked is the best split among a random subset of the features. The randomness causes the bias of the forest to slightly increase although due to averaging, its variance also decreases, usually compensating for the increase in bias, hence yielding a better model.

The main principle of AdaBoost (Adaptive Boosting) [7] is to fit a sequence of weak learners (i.e., models that are only slightly better than random guessing, such as small decision trees) on unceasingly changed versions of the information. The final prediction is produced by all the predictions which are combined through a weighted majority vote (or sum). Gradient Tree Boosting [8] is a generalization of boosting to arbitrary differentiable loss functions and is an accurate and an. effective procedure that can be used for both classification and regression. Gradient Tree Boosting models are used in for a multitude of use cases. XGBoost [9] stands for eXtreme Gradient Boosting is an accurate and scalable implementation of gradient boosting machines and was designed and developed for the purpose of computational speed and model performance. The implementation of XGBoost offers several features which are advanced for improving the algorithm, fine tuning model and, computing environments.

III. METHODOLOGY

The following chapter deals with the methodology for models not using AutoML and model using AutoML

A. Models not using AutoML

The methodology using models other than AutoML include the following steps: Converting text features to numerical features using One Hot Encoding, data normalization, splitting of dataset into training and testing, balancing the unbalanced training dataset, training the classifier and inferring from results. Figure 1 shows the above steps diagrammatically. Each of the steps are explained in the implementation chapter.

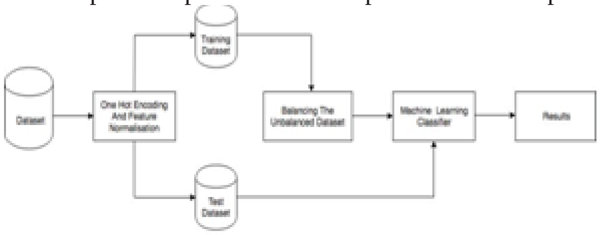
Training Dataset
Dataset
One Hot Encoding And Feature Normalisation
Balancing The Unbalanced Dataset
Machine Learning Classifier
Results
Test Dataset

Figure 1. Methodology of models not using AutoML

B. Models using AutoML

The methodology using the AutoML model include the following steps: Converting text features to numerical features using One Hot Encoding, training the classifier and inferring from results. The steps are significantly less complicated compared to models using AutoML. Figure 2 shows the above steps diagrammatically.

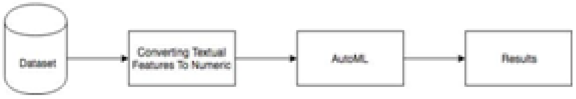

Figure 2. Methodology of model using AutoML

IV. IMPLEMENTATION

The following chapter deals with the implementation which includes the feature selection and feature engineering, the balancing of the unbalanced dataset and the experiment itself.

A. Feature Engineering And Selection

The dataset obtained from Kaggle for Employee Attrition contained 9 features out of which two of them were categorical features. The numeric features included number of projects the employee has undertaken, average monthly hours, time spent (in years) in the company, work related accidents (yes/no), promotion received in the last 5 years (yes/no), current employee satisfaction level on a scale of 0 to 1, previous employee satisfaction evaluation on a scale of 0 to 1. The categorical variables include division they work in (sales, accounting, human resources, support, technical, IT, management, product management, marketing, unspecified), the salary they receive (low, medium, high). The numerical features were normalized on a scale of 0 to 1. The categorical variables were converted to numerical variables using one hot encoding technique wherein each value in a single type of categorical feature is converted to a binary feature.

B. Balancing The Unbalanced Dataset

The dataset obtained was unbalanced with 3571 out of 14999 samples having the output of 1 which indicated they left the company while the rest of them had the result as 0 which meant they hadn’t left the company. This would lead to a biased prediction if no balancing was done and the model might seem falsely accurate after training and testing as a result of which model wouldn’t be suitable for real time use. For the purpose of balancing, initially all the samples were split into samples which contained employees who left and employees who hadn’t left. Around 800 samples from each set were taken and randomly mixed into a test set sample space. The rest of the samples from the set containing samples of employees who’d left the company were mixed with three different sets which were formed by splitting the remaining samples in the set of employees who hadn’t left as indicated in the Figure 3. For each classifier model, the three subsets formed were trained independently and for testing, each of the models were used on the test set and the results were aggregated by finding out which of the category (not left company/ left company) was predicted by a vote of majority for each sample.

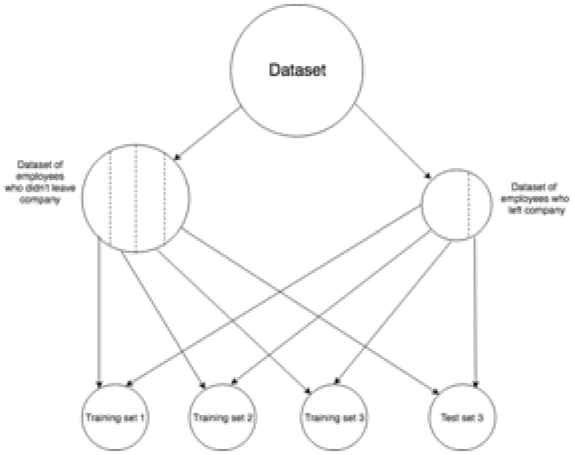
Dataset
Dataset of employees who didn't leave company
Dataset of employees who left company
Training set 1
Training set 2
Training set 3
Test set 3

Figure 3. Balancing the dataset

C. Experiment

The balancing of the dataset procedure as specified above was applied to each model apart from the AutoML model which has the characteristic of balancing the data implicitly. The various models used on the same splits of the data as specified in the balancing procedure were used on multiple models namely Support Vector Classifier with an

RBF Kernel, Logistic Regression, Stochastic Gradient Descent Classifier, Bagging Classifier, Random Forest Classifier, Ada Boost (Adaptive Boosting) Classifier, Gradient Boosting Classifier, XGBoost (eXtreme Gradient Boosting) Classifier. The results were compared for each of the above models in terms of accuracy and time efficiency. The Bagging Classifier, Random Forest Classifier, Ada Boost Classifier, Gradient Boosting Classifier, XGBoost Classifier were trained using 200 estimators

The experiment was carried out in Python 3.6.4 on macOS High Sierra v10.13.3, a 2.8 GHz Intel Core i7 processor, 16 GB 2133 MHz LPDDR3 RAM. The results obtained with respect to the time efficiency are based on the specifications above.

V. RESULT AND ANALYSIS

This chapter details the results and analysis of the experiments carried out. The results of the models with specific parameters are specified in the table 1. The models are compared on the basis of the accuracy in percent achieved on the test dataset and the time efficiency in seconds which is the time taken in seconds to train the model on the training dataset.

Table 1. results and analysis

Model Name	**Accuracy (In Percent)**	**Time Efficiency (In Seconds)**
AutoML	98.56	116.22655
Support Vector Classifier (RBF Kernel)	67.94	4.857004
Logistic Regression	67.12	0.083944
Stochastic Gradient Descent Classifier (1000 Maximum Iterations)	69.35	1.461704
Bagging Classifier (200 Estimators)	99.23	6.813611
Random Forest Classifier (200 Estimators)	99.17	2.565433
Ada Boost Classifier (200	94.00	2.098886

Estimators)		
Gradient Boosting Classifier (200 Estimators)	95.47	2.329223
XGBoost Classifier (200 Estimators)	98.88	3.790126

The figure 4 shows a scatter plot of Accuracy percent vs Time in seconds for each of the models. The scatter plot along with table 1 give conclusive insights into the utility of these models for this use case. The figure 5 indicates the relative feature importance in the dataset using F score. Inferring from the figure 5, the following factors majorly indicate the reason for the classification in descending order: average monthly hours, current employee satisfaction score (on a scale of 0-1), previous evaluation of satisfaction, number of projects been a part of, the number of years in the company.

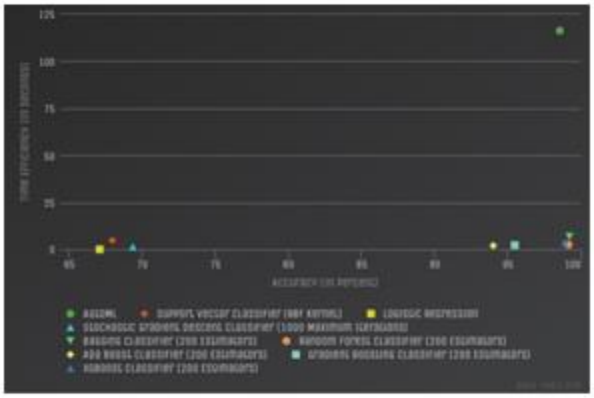

Figure 4: Accuracy percent vs Time in seconds

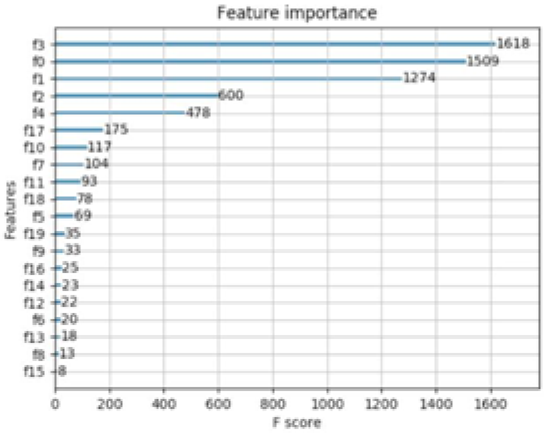

Figure 5: Feature importance graph. Features vs F score

VI. CONCLUSION

It was concluded that while AutoML was fairly accurate with over 98% accuracy, the time taken for training is much higher than the other models. However, AutoML which balances the unbalanced data requires minimal feature engineering and less expertise in feature engineering if one wants to use the dataset directly. The models such as Support Vector Classifier with RBF kernel, Logistic Regression, Stochastic Gradient Descent were accurate up to a percent of 68% which wasn't satisfactory for this use case. Ada Boost and Gradient Boosting classifier gave an accuracy of up to 95% with similar time of training. Random Forest Classifier, Bagging Classifier and XGBoost Classifier gave an accuracy of up to 99%. Bagging Classifier takes more time when compares to Random Forest Classifier and XGBoost Classifier. The major factors influencing the attrition were as follows: (in descending order) average monthly hours, current employee satisfaction score (on a scale of 0-1), previous evaluation of satisfaction, number of projects been a part of, the number of years in the company

VII. FUTURE WORK

The same approach will be extended to a larger dataset with more features which will potentially give a deeper and a more comprehensive insight into employee attrition. The models can be used in real time for actually gaining insight into the factors responsible for employee attrition and predicting employees who might leave potentially, and it can be prevented.

REFERENCES

1. *Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Springenberg ,Manuel Blum, Frank Hutter. Efficient and Robust Automated Machine Learning, Neural Information Processing Systems (NIPS),2015*
2. *Chih-Chung Chang and Chih-Jen Lin, LIBSVM: A Library for Support Vector Machines, 2001*
3. *Chao-ying Joanne Peng, Kuk Lida Lee, Gary M. Ingersoll, An Introduction to Logistic Regression Analysis and Reporting, The Journal of Educational Research, 2002*
4. *Yves Lechevallier and Gilbert Saporta, Large-Scale Machine Learning with Stochastic Gradient Descent, Proceedings of the 19th International Conference on Computational Statistics (COMPSTAT'2010), 177–187, 2010*
5. *Breiman, L, Bagging predictors, Machine Learning 24(2): 123-140, Online ISSN: 1573-0565, 1996*
6. *Breiman, L., Random Forests, Machine Learning 45(1): 5-32, Online ISSN: 1573-0565,2001*
7. *Yoav Freund, Robert E Schapire, A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting, Journal of Computer and System Sciences,Volume 55, Issue 1, Pages 119-139, ISSN 0022-0000, 1997 M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.*
8. *Friedman, Jerome H. Greedy function approximation: A gradient boosting machine. Ann. Statist. 29 no. 5, 1189--1232. doi:10.1214/aos/1013203451,2001*
9. *Tianqi Chen, Carlos Guestrin, XGBoost: Reliable Large-scale Tree Boosting System, Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016.*